



**ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ
ГОУ ВПО НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
им. Р.Е. АЛЕКСЕЕВА**

**ИНСТИТУТ РАДИОЭЛЕКТРОНИКИ И ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ**

Кафедра "Вычислительные системы и технологии"

ПРОГРАММИРОВАНИЕ

**Отчёты
о выполнении лабораторного практикума**

Выполнил студент группы 22-ИВТ-2 Лисенков Кирилл
Алексеевич

_____ «__» _____ 20__ г.
(личная подпись) (дата)

Провел старший преподаватель кафедры
«Вычислительные системы и технологии»
Мартынов Дмитрий Сергеевич

_____ «__» _____ 20__ г.
(личная подпись) (дата)

НИЖНИЙ НОВГОРОД 2023

1. Разрабатываемая программа предназначена для обработки и хранения массива экземпляров класса.

```
#ifndef LIB_H
#define LIB_H
#include <fstream>
#include <ios>

class Travel{
private:
    char name[20];
    char country[20];
    double price;
    double duration;
    char transport[20];
public:
    Travel();
    void setName(const char* newName);
    void setCountry(const char* newCountry);
    void setPrice(const double newPrice);
    void setDuration(const double newDuration);
    void setTransport(const char* newTransport);
    const char* getName()const;
    const char* getCountry()const;
    const double getPrice()const;
    const double getDuration()const;
    const char* getTransport()const;
    friend std::istream& operator>>(std::istream&, Travel&);
    friend std::ostream& operator<<(std::ostream&, const Travel&);
    void infile(std::ofstream& file);
};

#endif

#include "lib.h"
#include <iostream>
#include <fstream>
#include <iomanip>
#include <stdexcept>
#include <string.h>
#include <string>
#include <ios>
Travel::Travel(){}
void Travel::setName(const char* newName){strncpy(name, newName, sizeof(name));};

void Travel::setCountry(const char* newCountry){strncpy(country, newCountry,
sizeof(country));};

void Travel::setPrice(const double newPrice){price = newPrice;};

void Travel::setDuration(const double newDuration){duration = newDuration;};

void Travel::setTransport(const char* newTransport){strncpy(transport, newTransport,
sizeof(transport));};
```

```

const char* Travel::getName()const {return name;};

const char* Travel::getCountry()const {return country;};

const double Travel::getPrice()const {return price;};

const double Travel::getDuration()const {return duration;};

const char* Travel::getTransport()const {return transport;};


std::istream& operator>>(std::istream& is, Travel& Travel){
    static int counter = 1;
    std::cout << "Запись: " << counter << ":\n";
    std::cout << "    Наименование: ";
    if(!(is >> Travel.name)){
        throw std::runtime_error("Ошибка Ввода");
    }
    std::cout << "    Страна: ";
    if(!(is >> Travel.country)){
        throw std::runtime_error("Ошибка Ввода");
    }
    std::cout << "    Стоимость: ";
    if(!(is >> Travel.price)){
        throw std::runtime_error("Ошибка Ввода");
    }
    std::cout << "    Продолжительность: ";
    if(!(is >> Travel.duration)){
        throw std::runtime_error("Ошибка Ввода");
    }
    std::cout << "    Вид транспорта: ";
    if(!(is >> Travel.transport)){
        throw std::runtime_error("Ошибка Ввода");
    }
    counter++;
    return is;
}


std::ostream& operator<<(std::ostream& os, const Travel& Travel){
    static bool first = true;
    if (first){
        os << "-----\n" << std::left << std::setw(20) << "|Name" << std::setw(20) << "|Country" << std::setw(10) <<
"|Price" << std::setw(10) << "|Duration" << std::setw(20) << "|Transport|" << std::endl << "----\n";
        first = false;
    }
    os << std::left << std::setw(20) << Travel.name << std::left << std::setw(20) <<
Travel.country << std::setw(10) << std::fixed << std::setprecision(2) << Travel.price <<
std::setw(10) << std::fixed << std::setprecision(2) << Travel.duration << std::setw(20) <<
Travel.transport << '\n';
    return os;
}

```

```

void Travel::infile(std::ofstream& file){
    static bool first = true;
    if (first){
        file << "-----\n" << std::left << std::setw(20) << "|Name" << std::setw(20) << "|Country" << std::setw(10)
<< "|Price" << std::setw(10) << "|Duration" << std::setw(20) << "|Transport|" << std::endl <<
"-----\n";
        first = false;
    }
    file << std::left << std::setw(20) << name << std::left << std::setw(20) << country <<
std::setw(10) << std::fixed << std::setprecision(2) << price << std::setw(10) << std::fixed <<
std::setprecision(2) << duration << std::setw(20) << transport << '\n';
    return;
}

// figure: task1
//*****
// Filename: task1v19main.cpp
// Abstract: Простая программа на C++
// Description: программа предназначена для хранения и обработки массива переменных
структурного типа данных
// Create Date: 2023 / 04 / 05
// Author: Студент ИРИТ Лисенков К.А. 22-ИБТ-2
// Notes / Platform / Copyright IRIT NNTU/ UNIX/Linux / FreeWare
//*****

#include "lib.h"
#include <iostream>
#include <cstring>
#include <iomanip>
#include <fstream>
#include <ios>
#include <string>

int main(int argc, char** argv)
{
    std::cout << "*****\n" //Приветствие.
    << "* Нижегородский государственный технический университет *\n"
    << "* Лабораторная работа 3. Задание 1. Вариант 19 *\n"
    << "* Выполнил студент группы 22-ИБТ-2. Лисенков Кирилл *\n"
    << "*****\n";
    if((argc == 2) && ((strcmp(argv[1], "-h") == 0) || (strcmp(argv[1], "--help") == 0))){
//Получения справки
        std::cout << "Справка:\n"
        << "Добро пожаловать в программу для для хранения и обработки массива
переменных структурного типа данных.\n"
        << " -c [N] [file_name] - запуск программы в режиме создания электронной
таблицы записей, N – количество записей, file_name – имя текстового файла, в котором будет
сохранен массив (таблица) записей.\n"
        << " -r [N] [file_name] - запуск программы в режиме чтения содержимого
текстового файла file_name, на экран должны быть выведены не более N записей.\n";
        exit(0);
    }
    else if((argc == 4) && ((strcmp(argv[1], "-c") == 0) || (strcmp(argv[2], "--create") == 0))){

```

```

    if(atoi(argv[2])< 1){
        std::cout << "Ошибка, число записей не может быть отрицательным или равным нулю!";
        exit(-1);
    }
    size_t n = atoi(argv[2]);
    Travel* table = new Travel[n];
    std::ofstream file(argv[3]);
    std::ofstream &rfile = file;
    if(!file.is_open()){
        std::cout << "Ошибка открытия файла!";
        exit(-1);
    }
    file << "Общее число записей: " << n << std::endl;
    for(int i{0}; i < n; i++){
        std::cin >> table[i];
        table[i].infile(rfile);
    }
    for(int i{0}; i < n; i++){
        std::cout << table[i];
    }
    file.close();
    delete[] table;
}
else if((argc == 4) && ((strcmp(argv[1], "-r") == 0) || (strcmp(argv[2], "--read") == 0))){
//Чтения бд
    if(atoi(argv[2])< 1){
        std::cout << "Ошибка, число записей не может быть отрицательным или равным нулю!";
        exit(-1);
    }
    size_t n = atoi(argv[2]);
    std::ifstream file(argv[3]);
    if(!file.is_open()){
        std::cout << "Ошибка открытия файла!";
        exit(-1);
    }
    std::string line;
    int i{0};
    while (std::getline(file, line)){
        std::cout << line << std::endl;
        i++;
        if(i == n+4)
            break;
    }
}
}
}

```

Алгоритм "Хранение и обработка массива переменных структурного типа данных"

Начало

| Выводим приветствие

| Если аргументы командной строки равны "-h" или "--help"

```
|| Выводим справку по использованию программы
|| Завершаем программу
| Конец условия для получения справки
| Если аргументы командной строки равны "-с" или "--create"
|| Если второй аргумент N меньше 1
||| Выводим сообщение об ошибке "Ошибка, число записей не может быть отрицательным или равным нулю!"
||| Завершаем программу с кодом -1
|| Конец условия для проверки N
|| Преобразуем третий аргумент в тип size_t и присваиваем его значение переменной n
|| Создаем динамический массив table размером n типа Travel
|| Открываем файл с именем, указанным в четвертом аргументе, для записи и присваиваем его ссылку переменной file
|| Если не удалось открыть файл
||| Выводим сообщение об ошибке "Ошибка открытия файла!"
||| Завершаем программу с кодом -1
|| Конец условия для проверки открытия файла
|| Записываем общее число записей n в файл
|| Цикл для каждого i от 0 до n-1
||| Выводим сообщение "Запись i+1:"
||| Считываем данные от пользователя и сохраняем их в table[i]
||| Записываем данные в файл с использованием метода infile объекта table[i]
|| Конец цикла для i
|| Цикл для каждого i от 0 до n-1
||| Выводим данные из table[i]
|| Конец цикла для i
|| Закрываем файл
|| Освобождаем память, выделенную для массива table
| Конец условия для создания таблицы записей
| Если аргументы командной строки равны "-r" или "--read"
|| Если второй аргумент N меньше 1
```

||| Выводим сообщение об ошибке "Ошибка, число записей не может быть отрицательным или равным нулю!"

||| Завершаем программу с кодом -1

|| Конец условия для проверки N

|| Преобразуем третий аргумент в тип size_t и присваиваем его значение переменной n

|| Открываем файл с именем, указанным в четвертом аргументе, для чтения и присваиваем его ссылку переменной file

|| Если не удалось открыть файл

||| Выводим сообщение об ошибке "Ошибка открытия файла!"

||| Завершаем программу с кодом -1

|| Конец условия для проверки открытия файла

|| Читаем строки из файла и выводим на экран не более N записей

| Конец условия для чтения файла

Конец программы

```
~/Документы/University/2 semester/Programming/Labo3 ./App1 -c 2 file.txt
*****
* Нижегородский государственный технический университет *
* Лабораторная работа 3. Задание 1. Вариант 19 *
* Выполнил студент группы 22-ИВТ-2. Лисенков Кирилл *
*****
Запись: 1:
    Наименование: a
    Страна: a
    Стоимость: 1
    Продолжительность: 1
    Вид транспорта: a
Запись: 2:
    Наименование: b
    Страна: b
    Стоимость: 2
    Продолжительность: 2
    Вид транспорта: b

-----
|Name          |Country        |Price  |Duration |Transport|
-----
a              a              1.00   1.00    a
b              b              2.00   2.00    b
-----

~/Документы/University/2 semester/Programming/Labo3 ./App1 -h
*****
* Нижегородский государственный технический университет *
* Лабораторная работа 3. Задание 1. Вариант 19 *
* Выполнил студент группы 22-ИВТ-2. Лисенков Кирилл *
*****
Справка:
Добро пожаловать в программу для хранения и обработки массива переменных структурного типа данных.
-c [N] [file_name] - запуск программы в режиме создания электронной таблицы записей, N - количество записей, file_name - имя текстового файла, в котором будет сохранен массив (таблица) записей.
-r [N] [file_name] - запуск программы в режиме чтения содержимого текстового файла file_name, на экран должны быть выведены не более N записей.
```

~/Документы/University/2 semester/Programming/Labo3 ./App1 -r 1 file.txt

* Нижегородский государственный технический университет *

* Лабораторная работа 3. Задание 1. Вариант 19 *

* Выполнил студент группы 22-ИВТ-2. Лисенков Кирилл *

Общее число записей: 2

Name	Country	Price	Duration	Transport
a	a	1.00	1.00	a